

© EPODOC / EPO

PN - JP2000357119 A 20001226
PD - 2000-12-26
PR - JP19990169465 19990616
OPD - 1999-06-16
TI - DIRECTORY SERVICE SYSTEM
IN - KAWAKAMI NOBUHIKO;SHIGA KENTA;YUI HITOSHI;HIRASHIMA YOKO; KIKUCHI SATOSHI
PA - HITACHI LTD
IC - G06F12/00 ; G06F17/30

© WPI / DERWENT

TI - Directory service system of e-mail system, has server database in which information about one or more names other than user's identification name is stored in actual entry
PR - JP19990169465 19990616
PN - JP2000357119 A 20001226 DW200116 G06F12/00 013pp
PA - (HITA) HITACHI LTD
IC - G06F12/00 ;G06F17/30
AB - JP2000357119 NOVELTY - The directory server includes a database which stores information about user's identification name and attribute in an actual entry (1). The information about one or more another names (3) is maintained in the actual entry. The server searches or updates information about entry in database based on demand from client through a network.
- USE - For electronic mail system connected to computer networks such as LAN.
- ADVANTAGE - Implementation property of names other than user's identification name is improved along with enhanced search capability. Prevents mismatching of directory information during failure generation and hence reduces labor.
- DESCRIPTION OF DRAWING(S) - The figure shows the conceptual diagram of directory information.
- Actual entry 1
- Another name 3
- (Dwg.1/14)
OPD - 1999-06-16
AN - 2001-153803 [16]

© PAJ / JPO

PN - JP2000357119 A 20001226

This Page Blank (13pt)

- PD - 2000-12-26
- AP - JP19990169465 19990616
- IN - KIKUCHI SATOSHI;HIRASHIMA YOKOSHIGA KENTA;KAWAKAMI
NOBUHIKO;YUI HITOSHI
- PA - HITACHI LTD
- TI - DIRECTORY SERVICE SYSTEM
- AB - PROBLEM TO BE SOLVED: To improve a retrieval performance by
eliminating the need for a process for tracing between entries at the
time of retrieval by holding information regarding more than one
alias.
- SOLUTION: An actual entry1 representing a user holds as
properties a general name (cn), a mail address (mail), and an alias
name (alias Name)3 in addition to identification names DN:
cn=Sato, cu=Osaka, o=ABC, C=JP. As the alias name3, cn=Sato,
ou=Tokyo, o=ABC, c=JP which are aliases of an actual entry1 are
registered. A directory server having received a retrieval request to
retrieve the identification names DN: cn=Sato, ou=Osaka, o=ABC,
c=JP or cn=Sato, ou=Tokyo, o=ABC, c=JP regards the request as a
request as to the actual entry 1. This directory information holds an
alias in the actual entry 1 and does not generate an alias entry
having an entity. Namely, a virtual alias entry2 is obtained as
shown by the broken line.
- I - G06F12/00 ;G06F17/30

This page blank (copy)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-357119

(P2000-357119A)

(43) 公開日 平成12年12月26日 (2000. 12. 26)

(51) Int.Cl.⁷

G 0 6 F 12/00
17/30

識別記号

5 4 5

F I

G 0 6 F 12/00
15/40
15/403

テマコード* (参考)

5 4 5 A 5 B 0 7 5
3 1 0 F 5 B 0 8 2
3 2 0 C

審査請求 未請求 請求項の数 5 O L (全 13 頁)

(21) 出願番号

特願平11-169465

(22) 出願日

平成11年6月16日 (1999. 6. 16)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 菊地 聡

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 平島 陽子

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(74) 代理人 100095511

弁理士 有近 紳志郎

最終頁に続く

(54) 【発明の名称】 ディレクトリ・サービス・システム

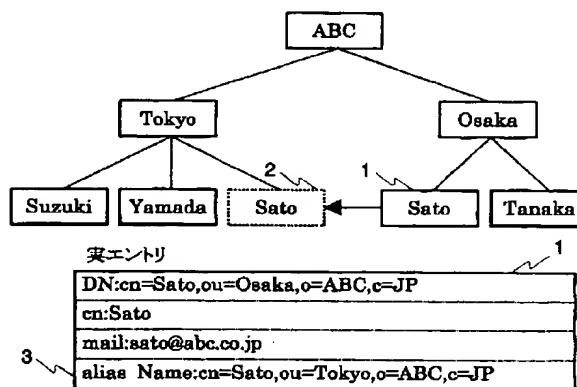
(57) 【要約】

【課題】 別名に関する運用性を向上させる。

【解決手段】 実エントリ1内に属性として別名3を保持し、従来のような実体を有するエイリアスエントリは作らない。エイリアスエントリは、破線で示すように、仮想のエイリアスエントリ2となる。

【効果】 別名に関する運用性に優れたディレクトリ・サービス・システムを提供できる。

図 1



【特許請求の範囲】

【請求項1】 ユーザの識別名と属性とを保持したエントリに関する情報を記憶するデータベースをディレクトリ・サーバに備えると共にネットワークを介してのクライアントからの要求に応じて前記ディレクトリ・サーバが前記データベース中のエントリに関する情報を検索または更新するディレクトリ・サービス・システムであって、前記エントリに、1以上の別名に関する情報を保持可能としたことを特徴とするディレクトリ・サービス・システム。

【請求項2】 請求項1に記載のディレクトリ・サービス・システムにおいて、前記エントリは、別名を属性として保持するか又は複数の識別名の一つとして保持することを特徴とするディレクトリ・サービス・システム。

【請求項3】 請求項1または請求項2に記載のディレクトリ・サービス・システムにおいて、前記ディレクトリ・サーバは、別名から対応するエントリに関する情報を検索または更新する別名制御手段を有することを特徴とするディレクトリ・サービス・システム。

【請求項4】 請求項1から請求項3に記載のディレクトリ・サービス・システムにおいて、エントリの別名を一元的に管理する一元管理手段を備えると共に、前記ディレクトリ・サーバは、更新する別名の一意性を前記一元管理手段に問い合わせる問合せ手段を有することを特徴とするディレクトリ・サービス・システム。

【請求項5】 請求項1から請求項4に記載のディレクトリ・サービス・システムにおいて、前記ディレクトリ・サーバは、前記クライアントからの単一の更新要求で1つのエントリと別名に対する更新とを受け付けることを特徴とするディレクトリ・サービス・システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ディレクトリ・サービス・システムに関し、さらに詳しくは、別名に関する運用性を向上させたディレクトリ・サービス・システムに関する。

【0002】

【従来の技術】PC (Personal Computer) 等の情報処理装置で作成した文書をLAN (Local Area Network) 等のネットワークを介して送受信する電子メールシステムの普及が進んでいる。

【0003】かかる電子メールシステムにおいて、受信者のメールアドレスを検索する、いわゆる電子電話帳機能として、CCITT勧告のX.500 (ISO9594) 等に代表されるディレクトリ・サービスが利用され始めている。X.500準拠のディレクトリ・サービスは、木構造 (ディレクトリツリー) として階層管理されたデータモデルを有する。木の枝葉に相当する個所には、エントリ (ディレクトリエントリ) が配置される。各々のエントリは、階層情報を含む識別名 (DN: Dist

inguished Name) で一意に識別され、ユーザのメールアドレス、姓名、電話番号、FAX番号、写真など様々な情報を属性として記憶可能である。X.500は、クライアント-サーバ型の分散システムアーキテクチャを採用している。クライアントおよびサーバの役割を担う情報処理装置間の通信プロトコルは、OSI (Open Systems Interconnection) の7レイヤ構造に従ったDAP (Directory Access Protocol) を規定している。

【0004】一方、インターネットにおける標準化機関であるIETF (Internet Engineering Task Force) は、TCP/IP上のディレクトリクライアント-サーバ間プロトコルとして「LDAP: Lightweight Directory Access Protocol (RFC2251)」を標準化した。

【0005】ユーザは、クライアント上のアプリケーションプログラムからX.500等のディレクトリ・サーバにDAPまたはLDAPでアクセスし、ユーザのメールアドレスなどの所望の情報を検索することが出来る。

【0006】また、DAPまたはLDAPは、エントリ追加、エントリ削除、エントリ名変更、属性値変更などのディレクトリ更新系要求も規定している。

【0007】ところで、X.500では、各ユーザの情報を格納するエントリ (実エントリと呼ぶ) の他に、実エントリの別名をエイリアスエントリで管理する別名機能について定義している。

【0008】従来のディレクトリ・サービスにおける別名機能は、例えば特開平7-129455号公報に記載されている。

【0009】図12は、従来のディレクトリ情報の例示図である。図12中、矩形はエントリであり、木構造を成す。35は、あるユーザを表す実エントリである。この実エントリ35は、識別名DN"cn=Sato, ou=Osaka, o=ABC, c=JP"に加えて、一般名 (cn) とメールアドレス (mail) を属性として保持している。36は、エイリアスエントリである。このエイリアスエントリ36は、識別名DN"cn=Sato, ou=Tokyo, o=ABC, c=JP"に加えて、実エントリ35の識別名DN"cn=Sato, ou=Osaka, o=ABC, c=JP"を被エイリアス名属性 (aliasedEntryName) 37として保持している。エイリアスエントリ36の識別名DN"cn=Sato, ou=Tokyo, o=ABC, c=JP"が、実エントリ35の別名となる。

【0010】ディレクトリ・サーバは、クライアントから受信した検索要求が実エントリを検索対象とする要求である場合は、該実エントリに含まれる属性値 (例えばメールアドレス) を検索結果として返送する。一方、ディレクトリ・サーバは、クライアントから受信した検索要求がエイリアスエントリを検索対象とする要求である場合は、該エイリアスエントリに含まれる被エイリアス名属性を参照し、該被エイリアス名属性に登録された識別名DNが表すエントリを検索対象に置換し、該置換したエントリに含まれる属性値 (例えばメールアドレス)

を検索結果として返送する。つまり、図12において、識別名DNが"cn=Sato,ou=Osaka,o=ABC,c=JP"である実エントリ35または識別名DNが"cn=Sato,ou=Tokyo,o=ABC,c=JP"であるエイリアスエントリ36を対象とした検索要求に対する結果は、何れも実エントリ35に含まれる属性値となる。

【0011】以上のように、ディレクトリ・サービスにおける別名機能は、複雑な組織構造をディレクトリ情報に反映させる上で有用な機能である。

【0012】

【発明が解決しようとする課題】上記従来技術では、次の問題点により、別名に関する運用性が良好ではなかった。

(1) クライアントから受信した検索要求が別名を検索対象とする要求であった場合、ディレクトリ・サーバは、エイリアスエントリを検索した後、実エントリを探索する必要があり、複数のエントリを辿る処理によるオーバーヘッドが発生し、検索性能が劣化しがちになる。

【0013】(2) エイリアスエントリも実体のあるディレクトリエントリであるため、エイリアスエントリに対する操作も実エントリに対する操作とは個別に行う必要があり、操作要求が多重になって操作に手間がかかる。また、処理に時間がかかるため、機器の電源断や異常などの不慮の事態により処理が中断され、ディレクトリ情報の整合性が損なわれてしまう可能性が高くなる。図13は、別名を持つエントリのエントリ追加、エントリ削除およびエントリ名変更時におけるクライアント・サーバ間の通信シーケンスの例示図である。エントリ追加の場合、クライアント5は、最初に実エントリを追加する。次に、登録した実エントリのエイリアスエントリを追加する。エントリ削除の場合、クライアント5は、最初に実エントリを削除する。次に、削除した実エントリのエイリアスエントリを削除する。エントリ名変更の場合、クライアント5は、最初に実エントリの識別名DNを変更する。次に、識別名DNを変更した実エントリのエイリアスエントリの被エイリアス名属性の値をエントリ名変更後の実エントリの識別名DNに置き換える。以上のように、操作に手間がかかる。更に、X.500では、1つの実エントリに複数の別名を付与することが許されているため、3以上の操作要求を必要とする場合がある。しかし、複数の更新要求を授受する間に機器の電源断や異常などの不慮の事態により処理が中断すると、ディレクトリ情報の整合性が損なわれてしまう。例えば、エントリ削除の処理が中断すると、実エントリは削除されているがエイリアスエントリは残存するといった状態が発生してしまう。

【0014】(3) エントリ削除またはエントリ名変更の場合、運用管理者は、操作対象の実エントリのエイリアスエントリの存否を常に意識しなければならない。しかし、実エントリの情報を参照するだけではエイリアス

エントリの存否を確認できない。このため、管理者は、実エントリの識別名DNが被エイリアス名属性に設定されているエイリアスエントリを検索するために、ディレクトリ・サーバ中の全エントリを探索しなければならず、手間がかかる。

【0015】(4) 付与できる別名が限定され、実エントリを管理するサーバの管理範囲以外のディレクトリ木に属する別名を付与できない。図14は、あるディレクトリ情報の例示図である。38は、実エントリであり、識別名DNは"cn=Sato,ou=Osaka,o=ABC,c=JP"である。39は、エイリアスエントリであり、被エイリアス名属性40に実エントリ38の識別名が保持されている。エイリアスエントリ39の識別名DN"cn=345,ou=People,o=ABC,c=JP"が、実エントリ38の別名となる。ここで、エントリ131(識別名DN"ou=People,o=ABC,c=JP")以下のエントリ群はサーバAに配置し、エントリ132(識別名DN"ou=Tokyo,o=ABC,c=JP")以下のエントリ群はサーバBに配置し、エントリ133(識別名DN"ou=Osaka,o=ABC,c=JP")以下のエントリ群はサーバCに配置する、という分散構成にした場合を想定する。人事異動により実エントリ38を変更する必要が生じたとき、サーバC上の実エントリ38を移動すると共にそのエイリアスエントリ39内の被エイリアス名属性40も変更しなければならない。ところが、セキュリティ等の問題から、サーバCを更新できるのはサーバCの運用管理者に限定されており、サーバAを更新できるのはサーバAの運用管理者に限定されているため、実エントリ38を管理するサーバCの運用管理者は、サーバAのエイリアスエントリ39を更新できない。従って、実エントリを保持するサーバの管理範囲以外のディレクトリ木に属する別名を付与できず、付与できる別名が限定されることになる。

【0016】(5) ある実エントリに複数の別名を付与する場合、相当する数のエイリアスエントリを追加しなければならない。

【0017】そこで、本発明の目的は、上記問題点

(1)～(5)を解消し、別名に関する運用性に優れたディレクトリ・サービス・システムを提供することにある。

【0018】

【課題を解決するための手段】第1の観点では、本発明は、ユーザの識別名と属性とを保持したエントリに関する情報を記憶するデータベースをディレクトリ・サーバに備えると共にネットワークを介してのクライアントからの要求に応じて前記ディレクトリ・サーバが前記データベース中のエントリに関する情報を検索または更新するディレクトリ・サービス・システムであって、前記エントリに、1以上の別名に関する情報を保持可能としたことを特徴とするディレクトリ・サービス・システムを提供する。上記第1の観点のディレクトリ・サービス・

システムでは、別名をエントリ自体に保持するため、検索時にエントリ間を辿る処理が不要となり、検索性能を向上できる。また、エントリの操作と別名の操作を一つの処理で達成でき、手間がかからず、障害発生時のディレクトリ情報の不整合を未然に防ぐことが出来る。また、エントリだけを参照することにより別名を容易に確認でき、別名を検索するための手間がかからない。また、付与できる別名が限定されず、エントリを管理する管理者の管理範囲以外のディレクトリ木に属する別名も付与できる。また、エントリに複数の別名を登録するだけで、エントリを増加させることなく、複数の別名を付与できる。

【0019】上記第1の観点のディレクトリ・サービス・システムにおいて、前記エントリは、別名を属性として保持するか又は複数の識別名の一つとして保持することが好ましい。また、上記第1の観点のディレクトリ・サービス・システムにおいて、前記ディレクトリ・サーバは、別名から対応するエントリに関する情報を検索または更新する別名制御手段を有することが好ましい。

【0020】第2の観点では、本発明は、上記第1の観点のディレクトリ・サービス・システムにおいて、エントリの別名を一元的に管理する一元管理手段を備えると共に、前記ディレクトリ・サーバは、更新する別名の一意性を前記一元管理手段に問い合わせる問合せ手段を有することを特徴とするディレクトリ・サービス・システムを提供する。上記第2の観点のディレクトリ・サービス・システムでは、他のサーバで登録された名称との重複を防止することが出来る。

【0021】第3の観点では、本発明は、上記第1または第2の観点のディレクトリ・サービス・システムにおいて、前記ディレクトリ・サーバは、前記クライアントからの単一の更新要求で1つのエントリと別名に対する更新とを受け付けることを特徴とするディレクトリ・サービス・システムを提供する。上記第3の観点のディレクトリ・サービス・システムでは、エントリの操作と別名の操作を一つの要求で達成でき、クライアントに負担がかからない。

【0022】

【発明の実施の形態】以下、本発明の実施の形態について図面を用いて説明する。図中、同一の部分には、同一の符号を付加する。

【0023】〔概要〕図1は、本発明に係るディレクトリ情報の概念図である。1は、あるユーザを表す実エントリである。この実エントリ1は、識別名DN"cn=Sato,ou=Osaka,o=ABC,c=JP"に加えて、一般名(cn)とメールアドレス(mail)とエイリアス名(aliasName)3とを属性として保持している。この外に、電話番号や住所などの様々な属性も追加可能である。前記エイリアス名3には、実エントリ1の別名である"cn=Sato,ou=Tokyo,o=ABC,c=JP"が登録されている。識別名DN"cn=Sato,ou=Osaka,o=ABC,c=JP"もしくは"cn=Sato,ou=Tokyo,o=ABC,c=JP"を検索対象とした検索要求を受けたディレクトリ・サーバは、どちらも実エントリ1を対象とする要求とみなす。

【0024】このように本発明に係るディレクトリ情報では、実エントリ1内に別名を保持し、従来例のような実体を有するエイリアスエントリは作らない(破線で示すように、仮想のエイリアスエントリ2となる)。

【0025】なお、本発明に係るエイリアス名3は別名の識別名DNを保持するための属性である。そして、別名を実エントリ中の一属性としているため、1回の操作により、別名をもつエントリのエントリ追加、エントリ削除、エントリ名変更を実現できる。これに対して、図12に示した従来のエイリアスエントリ36における被エイリアス名属性37は、実エントリ35の識別名DNを保持するための属性である。そして、エイリアスエントリも実エントリであるため、1回の操作では、別名をもつエントリのエントリ追加、エントリ削除、エントリ名変更を実現できない。

【0026】〔構成〕図2は、本発明に係るディレクトリ・サービス・システムの機能構成図である。このディレクトリ・サービス・システム100は、ディレクトリ・サーバ4とクライアント5とをネットワーク12で接続して構成されている。前記ディレクトリ・サーバ4は、ディレクトリエントリ情報を記憶するDB6と、クライアント5との間で通信処理を実行する通信制御部11と、クライアント5から受信したディレクトリアクセス要求を解析するプロトコル解析部10と、解析された要求に従って前記DB6を検索し要求を実行するエントリ管理部7とを具備して成る。前記エントリ管理部7は、実エントリに対するDBアクセス処理を実行する実エントリ管理部8と、別名に関わるDBアクセス処理を実行する別名制御部9とを具備して成る。前記クライアント5は、従来と同じ機能構成である。前記ネットワーク12は、LAN等である。

【0027】図3は、前記ディレクトリ・サーバ4のシステム構成図である。このディレクトリ・サーバ4は、CPU13と、磁気ディスク16と、主メモリ15と、キーボード21と、バス14と、ディスプレイ19と、マウス20とを具備して構成される。前記主メモリ15には、通信制御プログラム11と、プロトコル解析プログラム10と、実エントリ管理プログラム8および別名制御プログラム9を含むエントリ管理プログラム7とが格納されている。これらのプログラムは、当初磁気ディスク16に格納され、必要に応じて主メモリ15に転送された後、CPU13で実行される。前記磁気ディスク16には、ディレクトリDB6がファイルとして格納されている。このディレクトリDB6は、各種属性を含むディレクトリエントリ情報を格納するエントリ・テーブル17と、識別名DNを高速検索するためのインデック

ス情報を格納するDNインデックス・テーブル18とから成っている。前記クライアント5は、従来と同じシステム構成である。

【0028】[DB6のデータ構造]図4は、前記エン트리・テーブル17のデータ構造例示図である。このエン트리・テーブル17は、二次元の表形式を成し、各ディレクトリエントリ毎に一行が割り当てられる。また、各行には、ユニークな行識別番号が割り当てられ、エン트리ID22に格納される。また、一般名23、メールアドレス24、エイリアス名25などの各属性は、列に格納される。

【0029】図5は、前記DNインデックス・テーブル18のデータ構造例示図である。このDNインデックス・テーブル18は、二次元の表形式を成し、各識別名DNおよびエイリアス名毎に一行が割り当てられ、識別名DNまたはエイリアス名がDN26に格納される。また、各行に対応する前記エン트리・テーブル17の行識別番号がID27に格納される。従って、ある識別名DNとその別名とは、同一の行識別番号をID27に持つことになる。

【0030】[操作例および通信シーケンス]図6は、エン트리追加時の画面表示例である。エン트리追加時画面28は、追加する実エントリの識別名DNを入力するDN領域29、一般名cnを入力するcn領域30、メールアドレスを入力するmail領域31、別名を入力する別名領域32、入力した情報を設定するためのOKボタン33、入力中の情報をキャンセルするためのキャンセルボタン34から成る。ユーザは、クライアント5のキーボード等を用いてDN領域29～別名領域32に文字列を入力した後、OKボタン33をマウスでクリックする。

【0031】図7は、別名を持つエントリのエン트리追加、エン트리削除およびエン트리名変更時におけるクライアント-サーバ間の通信シーケンスの例示図である。各通信要求は既定のディレクトリアクセスプロトコル（例えばDAPまたはLDAP）である。エン트리追加の場合、クライアント5は、エイリアス名属性を含む実エントリを登録する。エン트리削除の場合、クライアント5は、エイリアス名属性を含む実エントリを削除する。従って、同時に別名も削除されるエン트리名変更の場合、クライアント5は、実エントリの識別名DNを変更する。

【0032】[動作]ディレクトリ・サーバ4は、クライアント5が発行した各種ディレクトリアクセス要求を通信制御部11で受信し、プロトコル解析部10でアクセス内容を解析した後、エントリ管理部7に対してエントリ更新（追加、変更）、エントリ削除、エントリ名変更、エントリ検索処理を依頼する。

【0033】図8は、エントリ管理部7のエントリ更新処理に関わる動作を表すフローチャートである。S40

1では、クライアント5からエントリ追加要求またはエントリ変更要求を受信すると、実エントリ管理部8は、該エントリの存否やアクセス権限等を確認し、当該更新要求を受け入れ可能か判定し、受け入れ可能ならば、別名制御部9に対して、別名に関わる処理を依頼する（S402へ移行する）。受け入れ不能ならば、S409へ移行する。S402では、別名制御部9は、別名属性値を含むエントリ追加要求である場合もしくは更新する別名属性値を含むエントリ変更要求である場合はS403へ移行し、別名属性に関わらない要求である場合はS405へ移行する。S403では、別名制御部9は、要求に含まれる別名属性値が一意（ユニーク）つまり未登録であるか否かをDNインデックス・テーブル18の検索によりチェックし、一意である場合はS404へ移行し、一意でない場合はS409へ移行する。S404では、別名制御部9は、要求に含まれる別名属性値でDNインデックス・テーブル18を更新する。すなわち、別名属性値を含むエントリ追加要求の場合、DNインデックス・テーブル18に新たな行を追加し、要求に含まれる別名をDN26に登録する。一方、更新する別名属性値を含むエントリ変更要求の場合は、DNインデックス・テーブル18から変更前の別名に相当する行を検索し、DN26を変更後の別名に置換する。

【0034】S405では、実エントリ管理部8は、当該更新要求の内容をDB6に反映する。すなわち、エントリ追加要求の場合は、エン트리・テーブル17に新たな行を追加し、各属性を対応する列に登録する。一方、エントリ変更要求の場合は、エン트리・テーブル17から変更対象のエントリに相当する行を検索し、変更対象の属性に相当する列に登録された値を要求に含まれる属性値に置換する。S406では、エントリ追加要求の場合はS407へ移行し、エントリ変更要求の場合はS408へ移行する。S407では、実エントリ管理部8は、DNインデックス・テーブル18に新たな行を追加し、追加エントリの識別名DNをDN26に登録する。

【0035】S408では、実エントリ管理部8は、通信制御部11を介して、正常に更新要求を処理した旨をクライアント5へ返送する。そして、処理を終了する。

【0036】S409では、実エントリ管理部8は、通信制御部11を介して、更新要求がエラーになった旨をクライアント5へ返送する。そして、処理を終了する。

【0037】図9は、エントリ管理部7のエントリ削除処理に関わる動作を示すフローチャートである。S501では、クライアント5からエントリ削除要求を受信すると、実エントリ管理部8は、該エントリの存否やアクセス権限等を確認し、当該削除要求を受け入れ可能か判定し、受け入れ可能ならば、別名制御部9に対して、別名に関わる処理を依頼する（S502へ移行する）。受け入れ不能ならば、S507へ移行する。S502では、別名制御部9は、エン트리・テーブル17を探索す

ることにより削除対象のエントリを見つけ、当該エントリのエイリアス名25に値が登録されているか否かチェックし、登録されている場合はS503へ移行し、値が未登録の場合はS504へ移行する。S503では、エイリアス名25に登録されている別名をDN26に持つ行をDNインデックス・テーブル18から削除する。

【0038】S504では、実エントリ管理部8は、削除対象のエントリに相当する行をエントリ・テーブル17から削除する。S505では、実エントリ管理部8は、削除したエントリに相当する行をDNインデックス・テーブル18から削除する。S506では、実エントリ管理部8は、通信制御部11を介して、正常に削除要求を処理した旨をクライアント5へ返送する。そして、処理を終了する。

【0039】S507では、実エントリ管理部8は、通信制御部11を介して、削除要求がエラーになった旨をクライアント5へ返送する。そして、処理を終了する。

【0040】図10は、エントリ管理部7のエントリ名変更処理に関わる動作を示すフローチャートである。本処理は従来と同様であり、別名制御部9は機能しない。S601では、クライアント5からエントリ名変更要求を受信すると、実エントリ管理部8は、該エントリの存否やアクセス権限等を確認し、当該エントリ名変更要求を受け入れ可能か判定し、受け入れ可能ならばS602へ移行し、受け入れ不能ならばS604へ移行する。S602では、DNインデックス・テーブル18を探索することにより変更前の識別名DNが登録された行を見つけ、DN26に登録された値を指定された変更後の識別名DNに置換する。S603では、通信制御部11を介して、正常にエントリ名変更要求を処理した旨をクライアント5へ返送する。そして、処理を終了する。

【0041】S604では、実エントリ管理部8は、通信制御部11を介して、エントリ名変更要求がエラーになった旨をクライアント5へ返送する。そして、処理を終了する。

【0042】図11は、エントリ管理部7のエントリ検索処理に関わる動作を示すフローチャートである。本処理は従来と同様であり、別名制御部9は機能しない。S701では、クライアント5からエントリ検索要求を受信すると、実エントリ管理部8は、DNインデックス・テーブル18を探索して、検索対象の識別名DNが登録された行を見つける。S702では、同一のDNがDN26に登録された行を発見した場合には(S702)、当該行に登録されたID27を参照し、同一のIDがID22に登録された行をエントリ・テーブル17から見つける。次に、実エントリ管理部8は、検索要求で指定された0以上の属性を発見した行から抽出し(S703)、検索結果としてクライアント5へ返送する(S704)。

【0043】-他の実施形態-

上記実施形態では、クライアント5から受信したエントリ更新要求またはエントリ削除要求に既定のエイリアス名3が含まれている場合に別名に関わる処理を実行するものとした(図8のS402、図9のS502)。しかし、別名として扱う属性をサーバの運用管理者が定義できるようにしても良い。この場合、図8のS402または図9のS502において、別名制御部9は、定義された属性が要求に含まれている場合に別名に関わる処理を実行する。

【0044】また、上記実施形態では、実エントリの識別名DNおよび別名をDNインデックス・テーブル18に登録している。しかし、DNインデックス・テーブル18に別名を登録しなくてもよい。この場合、図11のS701において、まず、DNインデックス・テーブル18を検索し、見つからないときは続いてエントリ・テーブル17のエイリアス名25を探索すれば良い。

【0045】また、上記実施形態では、各エントリの別名を該エントリの属性として保持している。しかし、各エントリの別名を該エントリの他の識別名DNとして保持しても良い。この場合、各エントリが複数の識別名DNを保持できるようサーバを変更すればよい。

【0046】また、上記実施形態では、実エントリを保持するサーバの管理範囲以外のディレクトリ木に属する別名も付与できる。しかし、図12の例のような分散サーバ構成の場合、他のサーバで登録された名称と重複してしまうことがある。これを確実に防止するためには、各ディレクトリサーバに登録された別名を一元的にデータベースで管理する別名管理サーバをさらに設ければ良い。各ディレクトリサーバは、別名に関する更新要求を受信した際、前記別名管理サーバに別名更新の可否を問い合わせる。別名管理サーバは、データベースを参照し、問い合わせのあった別名の一意性を確認できたなら、該別名をデータベースに登録すると共に、更新可をディレクトリサーバに通知する。一方、該別名が既にデータベースに登録済みであった場合は、更新不可をディレクトリサーバに通知する。

【0047】

【発明の効果】本発明のディレクトリ・サービス・システムによれば、次の各効果が得られ、別名に関する運用性を向上することが出来る。

(1) 別名をエントリ自体に保持するため、検索時にエントリ間を辿る処理が不要となり、検索性能を向上できる。

(2) 更新時には、エントリの操作と別名の操作を一つの要求・一つの処理で達成でき、手間がかからず、障害発生時のディレクトリ情報の不整合を未然に防ぐことが出来る。

(3) エントリに付与された別名を、当該エントリだけを参照することにより容易に確認できる。すなわち、別名を検索するための手間がかからない。

(4) 別名をエントリ自体に格納するため、該エントリを管理するサーバへのアクセス権限さえ有していれば、所望の別名を付与可能である。また、他のサーバの管理範囲のディレクトリ木に属する別名も付与可能である。これを許容する場合には、他のサーバで登録された名称との重複を防止するため、各運用管理者間でネーミング規則を制定すれば良い。図14の例の場合、東京の運用管理者は"cn=10000,ou=People,o=ABC,c=JP"から"cn=19999,ou=People,o=ABC,c=JP"までの別名を各東京ユーザに割り当て、大阪の運用管理者は"cn=20000,ou=People,o=ABC,c=JP"から"cn=29999,ou=People,o=ABC,c=JP"までの別名を各大阪ユーザに割り当てるように予め取り決めておくことにより、名称の衝突を防ぐことができる。また、別名を一元的に管理するサービスをシステムに導入すれば、一意な名称を確実に付与可能である。

(5) エントリに複数の別名を登録するだけで、エントリを増加させることなく、複数の別名を付与できる。

【図面の簡単な説明】

【図1】本発明に係るディレクトリ情報の概念図である。

【図2】本発明に係るディレクトリ・サービス・システムの機能構成図である。

【図3】ディレクトリ・サーバのシステム構成図である。

【図4】エントリ・テーブルのデータ構造例示図である。

【図5】DNインデックス・テーブルのデータ構造例示図である。

【図6】エントリ追加時の画面表示例である。

【図7】別名を持つエントリのエントリ追加、エントリ削除およびエントリ名変更時におけるクライアントサーバ間の通信シーケンスの例示図である。

【図8】エントリ管理部のエントリ更新処理に関わる動作を表すフローチャートである。

【図9】エントリ管理部のエントリ削除処理に関わる動作を表すフローチャートである。

【図10】エントリ管理部のエントリ名変更処理に関わる動作を表すフローチャートである。

【図11】エントリ管理部のエントリ検索処理に関わる動作を表すフローチャートである。

【図12】従来のディレクトリ情報の例示図である。

【図13】従来の別名に関わるエントリのエントリ追加、エントリ削除およびエントリ名変更時におけるクライアントサーバ間の通信シーケンスの例示図である。

【図14】従来のあるディレクトリ情報の例示図である。

【符号の説明】

1…実エントリ

3…別名属性

4…ディレクトリ・サーバ

5…クライアント

6…DB

7…エントリ管理部

8…実エントリ管理部

9…別名制御部

10…プロトコル解析部

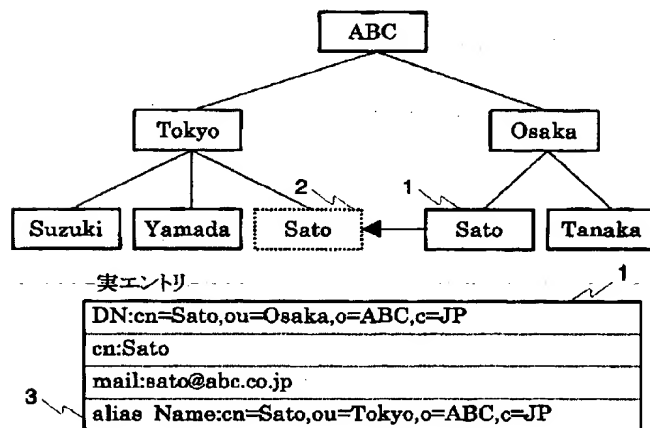
11…通信制御部

12…ネットワーク

100…ディレクトリ・サービス・システム

【図1】

図 1



【図5】

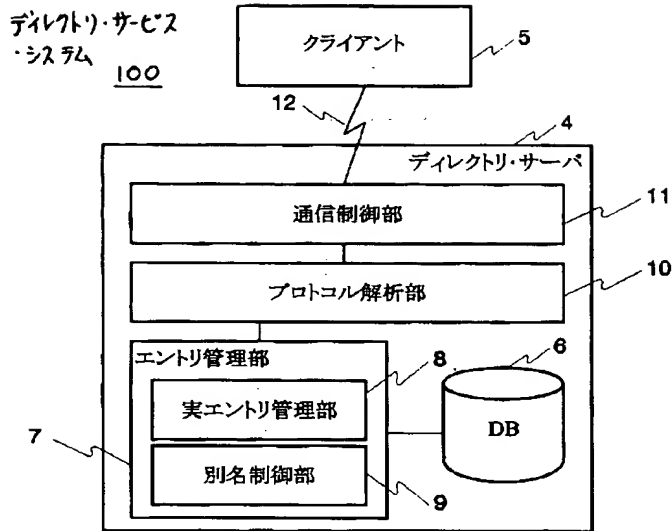
図 5

DN	ID
cn=Suzuki,ou=Tokyo,o=ABC,c=JP	1
cn=Yamada,ou=Tokyo,o=ABC,c=JP	2
cn=Sato,ou=Osaka,o=ABC,c=JP	3
cn=Tanaka,ou=Osaka,o=ABC,c=JP	4
cn=Sato,ou=Tokyo,o=ABC,c=JP	3

DNインデックス・テーブル

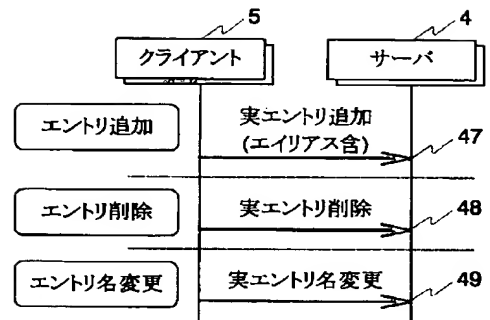
【図2】

図 2



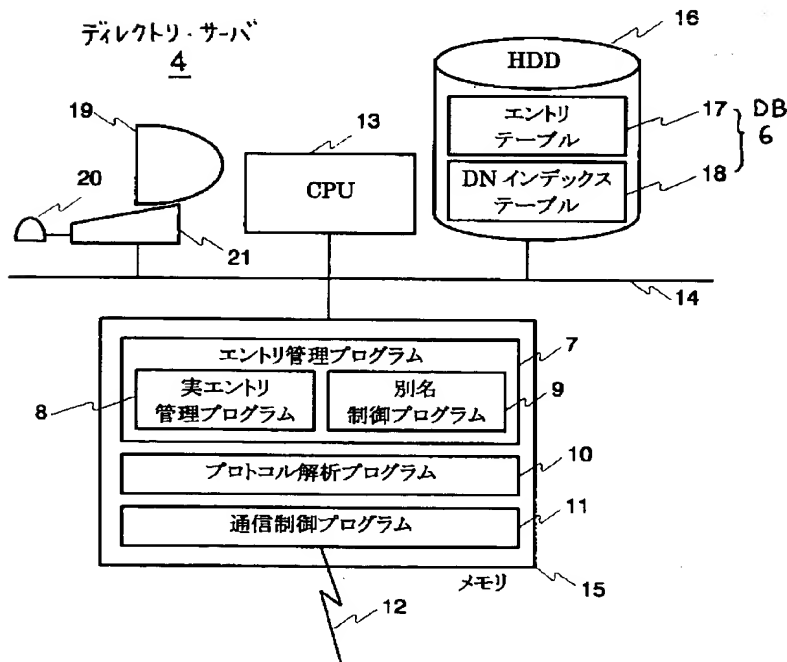
【図7】

図 7



【図3】

図 3



【図4】

図 4

ID	cn	mail	aliasName
1	Suzuki	suzuki@abc.co.jp	
2	Yamada	yamada@abc.co.jp	
3	Sato	sato@abc.co.jp	cn=Sato,ou=Tokyo,o=ABC,c=JP
4	Tanaka	tanaka@abc.co.jp	

エントリー・テーブル

【図6】

図 6

28

エントリー追加

29

エントリー名:

cn=Sato,ou=Osaka,o=ABC,c=JP

30

一般名:

Sato

31

メール・アドレス:

sato@abc.co.jp

32

別名:

cn=Sato,ou=Tokyo,o=ABC,c=JP

OK

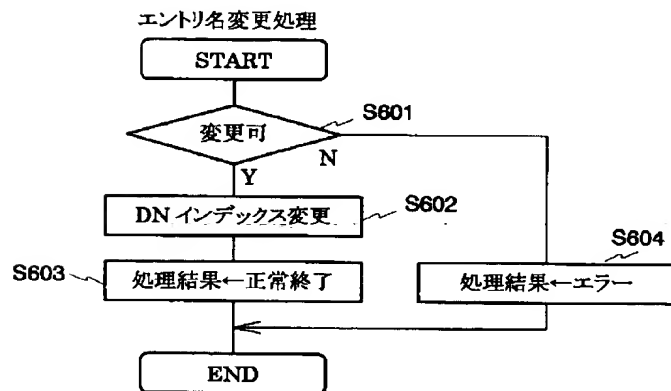
キャンセル

33

34

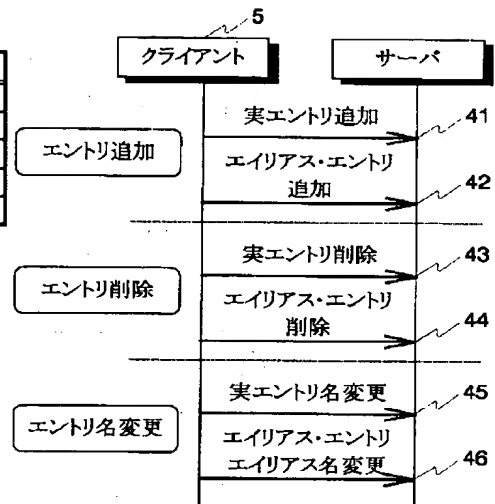
【図10】

図 1 0



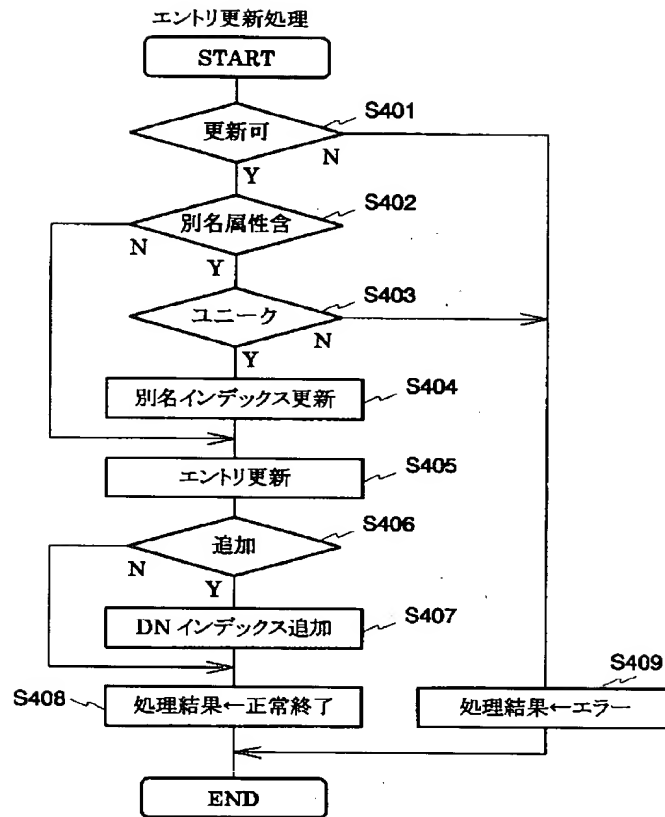
【図13】

図 1 3



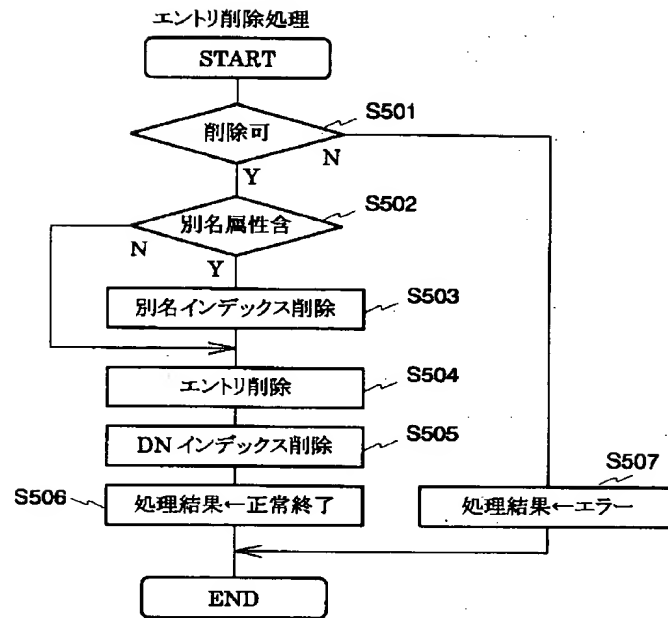
【図8】

図 8



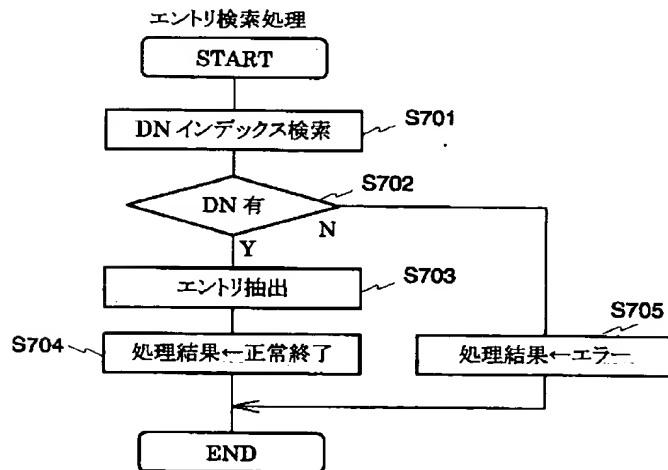
【図 9】

図 9



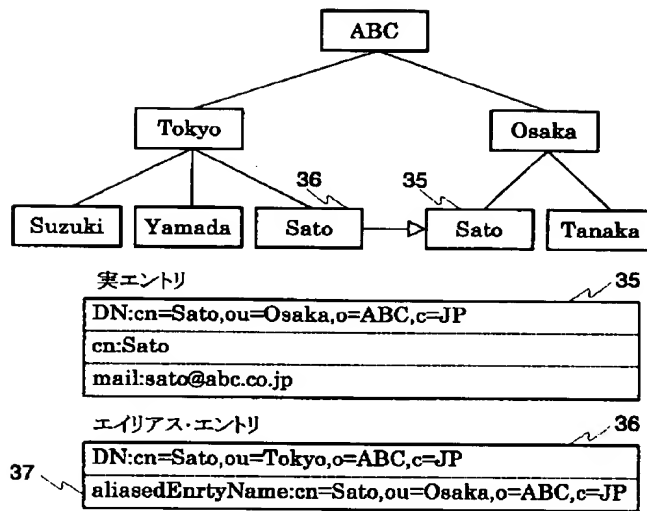
【図 11】

図 11



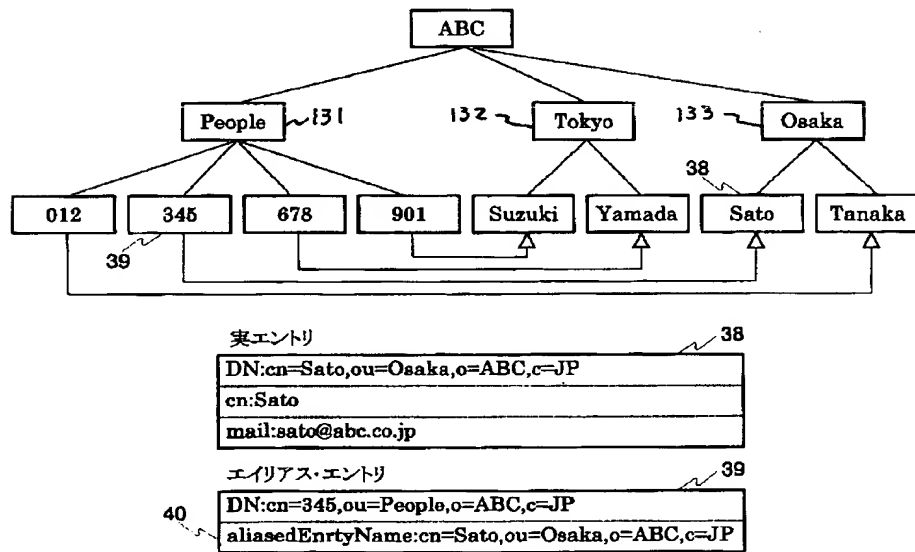
【図12】

図 1 2



【図14】

図 1 4



フロントページの続き

(72)発明者 志賀 賢太
神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内

(72)発明者 川上 順彦
神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内

(13) 000-357119 (P2000-3558)

(72) 発明者 由井 仁
神奈川県横浜市戸塚区5030番地 株式会社
日立製作所ソフトウェア事業部内

Fターム(参考) 5B075 NK43 NK46 PP02 PP03 PP12
5B082 GA08 HA00

This Page Blank (uspto)

This Page Blank (copy)